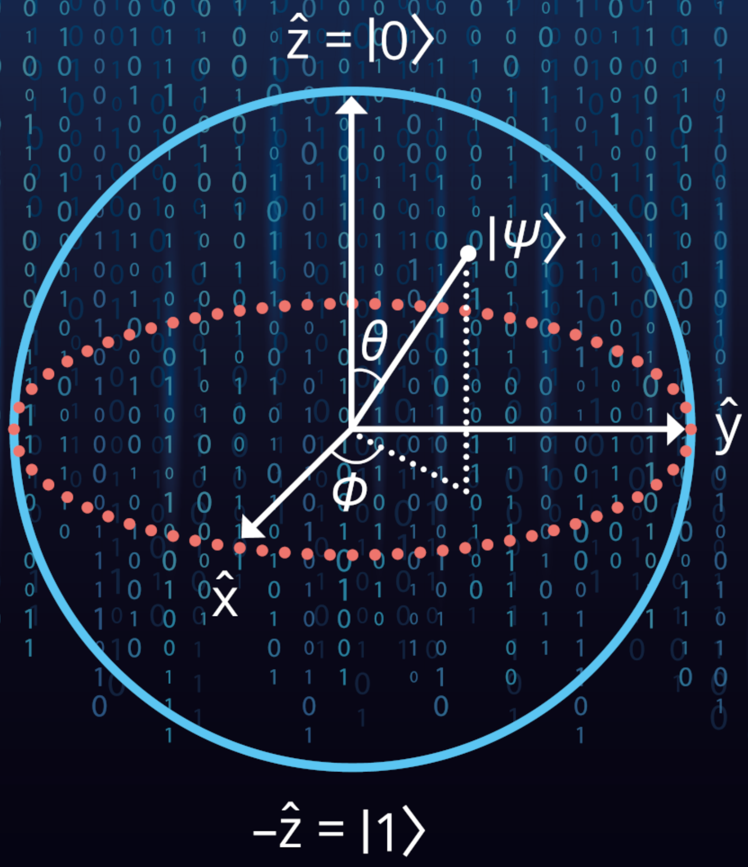
EUROPEAN UNION AGENCY
FOR CYBERSECURITY



POST-QUANTUM CRYPTOGRAPHY

Current state and quantum mitigation

MAY 2021, v2

ABOUT ENISA

The European Union Agency for Cybersecurity, ENISA, is the Union's agency dedicated to achieving a high common level of cybersecurity across Europe. Established in 2004 and strengthened by the EU Cybersecurity Act, the European Union Agency for Cybersecurity contributes to EU cyber policy, enhances the trustworthiness of ICT products, services and processes with cybersecurity certification schemes, cooperates with Member States and EU bodies, and helps Europe prepare for the cyber challenges of tomorrow. Through knowledge sharing, capacity building and awareness raising, the Agency works together with its key stakeholders to strengthen trust in the connected economy, to boost resilience of the Union's infrastructure, and, ultimately, to keep Europe's society and citizens digitally secure. For more information, visit www.enisa.europa.eu.

CONTACT

For contacting the authors please use evangelos.rekleitis@enisa.europa.eu
For media enquiries about this paper, please use press@enisa.europa.eu

AUTHORS

Ward Beullens,
Jan-Pieter D'Anvers,
Andreas Hülsing,
Tanja Lange,
Lorenz Panny,
Cyprien de Saint Guilhem,
Nigel P. Smart.

FOR ENISA

Evangelos Rekleitis,
Angeliki Aktypi,
Athanasios-Vasileios Grammatopoulos.

LEGAL NOTICE

COPYRIGHT NOTICE

ISBN: 978-92-9204-468-8, DOI: 10.2824/92307

EXECUTIVE SUMMARY

Quantum Technology is an emerging field of physics and engineering, which exploits the principles of quantum physics, like quantum entanglement, quantum superposition and quantum tunnelling, to provide new paradigms and novel applications. From computing and communications to metrology and imaging, research in the last 2 decades has bear tangible and not so tangible results. It is a critical technology that policy makers believe it will generate a multi-billion euro market in new technological solutions for business and citizens.

Since the beginning, the EU has been a key player in this area and with a planned investment of €1 billion over 10 years, the EU Quantum Flagship [1] is mobilising around 2000 scientists and industrialists, in a collaborative initiative on an unprecedented scale to position Europe as leader in the industrial landscape. Of course, Europe is not alone; the USA, China, Canada, and Japan have also set this as a top strategic priority.

Quantum Technology and in particular Quantum Computing is seen as a disruptive innovation. In the mid '90s, scientists theorized of quantum computer algorithms that, given the existence of a sufficiently powerful quantum computer, can break widely used public-key cryptography schemes, such as RSA and ECC or weaken standardised symmetric encryption algorithms. And while we do not know when and if such a quantum machine will [ever] become available, researchers and national authorities have been working on solutions. As a result, the US National Institute of Standards and Technology (NIST) launched in 2017 a, still ongoing, process to standardise one or more quantum-resistant public-key cryptographic algorithms, soliciting proposals from cryptographers around the world [2].

It is important to make a distinction between Post-Quantum Cryptography (PQC) and Quantum Cryptography. PQC is about designing cryptographic solutions that can be used by today's [non-quantum] computers and that we believe are resistant to both conventional and quantum cryptanalysis. On the other hand, Quantum Cryptography is about cryptographic solutions that take advantage of quantum physics to provide certain security services. Quantum Key Distribution (QKD) is a good example of the latter.

The EU Cybersecurity Strategy [3] , presented by the European Commission and the High Representative of the Union for Foreign Affairs and Security in Policy on December 2020, explicitly singles out quantum computing and encryption as a key technologies (along with AI) for achieving (1) resilience, technological sovereignty and leadership, (2) building operational capacity to prevent, deter and respond, and (3) advancing a global and open cyberspace. The Strategy covers the security of essential services such as hospitals, energy grids and railways and ever-increasing number of connected objects in our homes, offices and factories, building collective capabilities to respond to major cyberattacks and working with partners around the world to ensure international security and stability in cyberspace[4].

[1] https://qt.eu/. Last accessed May 2021.
[2] https://csrc.nist.gov/projects/post-quantum-cryptography. Last accessed May 2021.
[3] https://ec.europa.eu/digital-single-market/en/cybersecurity-strategy. Last accessed May 2021.
[4] https://ec.europa.eu/digital-single-market/en/cybersecurity. Last accessed May 2021.

Given recent developments in the Quantum Computing race among industries and nation states, it seems prudent for Europe to start considering mitigation strategies now. The European Union Agency for Cybersecurity is not alone in this line of thought. Other authorities and EU Institutions have also raised concerns; for instance, the European Data Protection Supervisor has highlighted the dangers against data protection [5] , national authorities have been investigating and preparing; e.g., the German Federal Office for Information Security has been evaluating Post-Quantum alternatives since before the launch of NIST's standardisation process [6].

This study provides an overview of the current state of play on the standardisation process of Post-Quantum Cryptography (PQC). It introduces a framework to analyse existing proposals, considering five (5) main families of PQC algorithms; viz. code-based, isogeny-based, hash-based, lattice-based and multivariate-based. It then goes on to describe the NIST Round 3 finalists for encryption and signature schemes, as well as the alternative candidate schemes. For which, key information on cryptodesign, implementation considerations, known cryptanalysis efforts, and advantages & disadvantage is provided.

Since the NIST standardisation process is going [7] , the report makes no claim on the superiority of one proposal against another. In most cases the safest transition strategy involves waiting for national authorities to standardise PQC algorithms and provide a transition path. There might be cases though, where the quantum risk in not tolerated, in which case the last chapter offers 2 proposals that system owners can implement now in order to protect the confidentiality of their data against a quantum capable attacker; namely hybrid implementations that use a combination of pre-quantum and post-quantum schemes, and the mixing of pre-shared keys into all keys established via public-key cryptography. These solutions come at a cost and as such system designers are well advised to perform a thorough risk and cost-benefit analysis.

The first version of this report was published in February 2021. The second version was released in May 2021 including a new chapter - Chapter 3 "Security Notions and Generic Transforms". The added material introduce key concepts of PQC cryptography, in order to make the report more self-contained.

[5] EDPS, "TechDispatch #2/2020: Quantum Computing and Cryptography", https://edps.europa.eu/data-protection/our-work/publications/techdispatch/techdispatch-22020-quantum-computing-and_en. Last accessed May 2021.

[6] https://www.bsi.bund.de/EN/Topics/Crypto/Cryptography/PostQuantumCryptography/post_quantum_cryptography_node.html. Last accessed May 2021.

[7] tentative deadline for Draft Standards is 2022/2024, as of 2021 https://csrc.nist.gov/projects/post-quantum-cryptography/workshops-and-timeline. Last accessed May 2021.

CONTENTS

1 Introduction 1

2 Families of Post-Quantum Algorithms 3
2.1 Code-based . . . 3
2.2 Isogeny-based . . . 3
2.3 Hash-based . . . 4
2.4 Lattice-based . . . 4
2.5 Multivariate-system based . . . 5
2.6 The NIST Round 3 Candidates . . . 5

3 Security Notions and Generic Transforms 7
3.1 PKE and KEMs . . . 7
3.1.1 Security Notions, and Transforms . . . 8
3.1.2 FO-transform . . . 9
3.2 Digital Signatures . . . 10
3.2.1 Identification schemes . . . 10
3.2.2 Trapdoor Functions . . . 11
3.2.3 One- and few-time signatures . . . 12
3.3 Quantum adversaries . . . 13
3.3.1 The quantum-accessible random oracle model . . . 13

4 NIST Round 3 Finalists 14
4.1 Encryption Schemes . . . 14
4.1.1 Classic McEliece . . . 14
4.1.2 Crystals-Kyber . . . 15
4.1.3 NTRU . . . 16
4.1.4 Saber . . . 17
4.2 Signature Schemes . . . 18
4.2.1 Crystals-Dilithium . . . 18
4.2.2 Falcon . . . 19
4.2.3 Rainbow . . . 20

5 Alternate Candidates 22
5.1 Encryption Schemes . . . 22
5.2 Signature Schemes . . . 23

6 Quantum Mitigation 25
6.1 Hybrid schemes . . . 25
6.2 Protective measures for pre-quantum cryptography . . . 26

7 Conclusions 28

Bibliography 29

1 INTRODUCTION

Post-quantum cryptography is an area of cryptography in which systems are studied under the security assumption that the attacker has a quantum computer. This attack model is interesting because Peter Shor has shown, since 1994, a quantum algorithm that breaks RSA, ECC, and finite field discrete logarithms in polynomial time. This means that, in this model, all commonly used public-key systems are no longer secure.

Symmetric cryptography is also affected, but significantly less. For systems that do not rely on mathematical structures the main effect is that an algorithm due to Lov Grover, from 1996, which halves the security level. This means that breaking AES-128 takes 2^{64} quantum operations, while current attacks take 2^{128} steps. While this is a big change, it can be managed quite easily by doubling the key sizes, e.g., by deploying AES-256. The operations needed in Grover's algorithm are inherently sequential which has led some to doubt that even 2^{64} quantum operations are feasible, but since the remedy of changing to larger key sizes is very inexpensive it is generally recommended to do so.

At this moment, the quantum computers that exist are not large enough to pose a threat against current cryptography. However, rolling out new cryptographic systems takes a lot of time and effort, and it is thus important to have replacements in place well before large, powerful quantum computers exist.

What makes matters worse is that any ciphertext intercepted by an attacker today can be decrypted by the attacker as soon as he has access to a large quantum computer (Retrospective decryption). Analysis of Advanced Persistent Threats (APT) and Nation State capabilities, along with whistle-blowers' revelations have shown that threat actors can and are casually recording all Internet traffic in their datacentres and that they select encrypted traffic as interesting and worth storing. This means that any data encrypted using any of the standard public-key systems today will need to be considered compromised once a quantum computer exists and there is no way to protect it retroactively, because a copy of the ciphertext is in the hands of the attacker. This means that data that needs to remain confidential after the arrival of quantum computers need to be encrypted with alternative means.

Signatures can be updated and old keys can be revoked when a signature system is broken; however, not all development in the area of building quantum computers is public and it is fairly likely that the first fully-functional large quantum computer will not be publicly announced, but rather sit in the basement of some government agency. Timing the roll-over of signature keys thus remains guesswork. On top of that, one important use case for signatures is operating-system upgrades. If a post-quantum signature system is not in place by the time an attacker has a quantum computer, then the attacker can take control of the operating system through a fake upgrade and prevent any future upgrades from fixing the problem.

In 2017, the United States National Institute for Standards and Technology solicited submissions for potential public key encryption and signature algorithms that would be secure in a world in which quantum computer existed. Although not officially a 'competition' as the AES and SHA-3 efforts were, it has been treated in much the same way as the AES and SHA-3 efforts. Over the last few years, the number of submissions has been whittled down, and in July 2020 the Round 3 candidates were published.

This report is a much extended update to the ECRYPT-CSA "Whitepaper on Post-Quantum Cryptography" [47]. It provides a short summary of the underlying hardness assumptions in Section 2 and summarizes the Round 3 candidates in Section 4. It also details the so-called 'Alternate Candidates' in Section 5. The Round 3 candidates are algorithms that the National Institute of Standards and Technology (NIST) *"considers to be the most promising to fit the majority of use cases and most likely to be ready for standardisation soon after the end of the third round"*, whilst the Alternate Candidates are ones which NIST regards as *"potential candidates for future standardisation, most likely after another round of evaluation"*. See [95] for more details. Finally, this report covers mitigation strategies in Section 6.

2 FAMILIES OF POST-QUANTUM ALGORITHMS

There would not be much point speaking about post-quantum systems, if there were none able to survive attacks by quantum computers. The usual disclaimers apply as with all of cryptography: It might be possible that more powerful attacks (quantum or not) exist that have not yet been found. Apart from that possibility, research over the last 15–20 years has built confidence in the following four areas that lead to secure systems in a post-quantum world. In this section, we summarize the mathematical basis of post-quantum proposals.

2.1 CODE-BASED

Code-based cryptography uses the theory of error-correcting codes. For some specially constructed codes it is possible to correct many errors, while for random linear codes this is a difficult problem. Code-based encryption systems go back to a proposal by McEliece from 1978 [86] and are among the most studied post-quantum schemes. Some code-based signature systems have been designed to offer short signatures at the expense of very large key sizes. Systems based on binary Goppa codes are generally considered secure; systems based on quasi-cyclic medium-density parity checks have held up to analysis for about a decade and are gaining confidence. For more background on code-based cryptography see [75].

All code-based signature systems submitted to NIST were based on new assumptions and have since been broken. Six code-based encryption systems made it to Round 2, but rank-metric codes (Rollo and RQC), as well as low-density parity-check (LDPC) codes (LEDAkem and LEDAcrypt) had serious cryptanalysis during Round 2 and were thus deselected by NIST.

The remaining code-based candidates are Classic McEliece, BIKE and HQC. Classic McEliece was the finalist selected first, for encryption systems; whereas BIKE and HQC were selected as alternate candidates. The latter two are using 'more' special codes in order to reduce the key size of the public key, which is seen as the main drawback of code-based systems.

2.2 ISOGENY-BASED

An isogeny between elliptic curves is a non-constant map that can be written as a fraction of polynomials and is compatible with addition on both curves, so that the image of the sum of two points on the first curve is equal to the sum of the images, when computed on the second curve. Isogeny-based cryptography uses isogenies between elliptic curves over finite fields. The isogeny problem is to find an isogeny between two elliptic curves that are known to be isogenous. The problem was introduced in 2005 in [30] and is thus the most recent basis for any post-quantum candidates. Usage in protocols differs in whether the degree of the isogeny is known or secret and whether additional information is known. For more background on isogeny-based cryptography see [76].

Only one isogeny-based candidate, SIKE, was submitted to the NIST competition and SIKE is in the third round as an alternate candidate.

2.3 HASH-BASED

Hash functions are functions that map strings of arbitrary length to strings of fixed length. From cryptographic hash-functions we expect that they are one-way (it is hard to find an element in the preimage of a given image) and collision resistant (it is hard to find two inputs that map to the same output). Hash functions are one of the most widely deployed cryptographic tools we got, with applications ranging from password hashing to file checksums, and are used in virtually any cryptographic construction in practice. While hash functions are used in all practical signature schemes to handle arbitrary length messages, it is known, since the beginning of public key cryptography, that they can also be used as the sole building block for this. In the simplest version, a hash-based signature on one bit is as follows. Pick two random strings, hash each of them, and publish the outputs. Reveal the first preimage to sign 0 and the second to sign 1. This signature scheme, due to Lamport from 1979 [74], is a one-time signature scheme – once the secret is revealed it cannot be used a second time. Starting from this basic idea hash-based signatures on longer strings and on multiple messages have been built. The designs fall into stateless and stateful versions. The former work as normal signatures, while for the latter the signer needs to keep track of some information, e.g., the number of signatures generated using a given key. With SPHINCS$^+$ a stateless hash-based signature scheme is in the third round of the competition as runner-up. For the stateful schemes, NIST already published SP 800-208 [32] standardizing LMS [87] and XMSS [59] two stateful hash-based signature schemes. However, it has to be noted that the stateful character limits the applications these schemes are suitable for.

Due to their ubiquity, the security of practical hash functions is well understood. More importantly in the given context, it is known that even quantum computers cannot significantly improve the complexity of generic attacks against cryptographic hash functions. A square-root factor speed-up is the (in practice unreachable) upper limit for improvements.

2.4 LATTICE-BASED

On a high level, the descriptions of lattices look much like those of codes – elements are length-n vectors in some space and get error vectors added to them – but where codes typically have entries 0 or 1, lattices work with much larger numbers in each entry and errors can move away further. The problems underlying the cryptographic constructions are to find the original vector given a disturbed one. Lattices offer more parameters than codes, which means that they might offer solutions better adapted to a given situation, but also offer more attack surface. Lattice-based cryptography goes back to 1996 and the designs of Ajtai [1] and of Hoffstein, Pipher, and Silverman [55]. Both encryption and signature systems exist.

The lattice based schemes submitted to NIST mainly make use of the following two basic hard problems; called Module-Learning-with-Errors (Module-LWE) and Module-Learning-with-Rounding (Module-LWR). In these schemes one selects a polynomial ring $R = \mathbb{Z}[X]/f$, where the degree of f is equal to n, and considers it modulo q (giving R_q). In addition, there is another integer parameter d, called the module degree. For Ring-LWE and Ring-LWR one sets $d = 1$, and for standard LWE and LWR one has $d = n = 1$.

The Module-LWE problem is the problem of finding $s \in R_q^d$ given a number of samples of the form $(a, a \cdot s + e)$ where a is chosen uniformly at random in R_q^d and $e \in R_q$ is chosen to have 'small' coefficients.

The Module-LWR problem is the problem of finding $s \in R_q^d$ given a number of samples of the form $(a, \lfloor a \cdot s \rceil_p)$ where a is chosen uniformly at random in R_q^d, and the function $\lfloor g \rceil_p$ takes the coefficients of the polynomial g and applies the function $x \longmapsto \text{round} - \text{to} - \text{int}(x \cdot p/q) \pmod p$, for some fixed integer p.

A related hard problem is that of the NTRU problem. NTRU-based cryptosystems, also called Quotient NTRU cryptosystems, assume that the NTRU problem is hard and that the n-sample Ring-LWE problem is hard, while Ring-LWE-based cryptosystems assume that the $2n$-sample Ring-LWE problem is hard. The NTRU problem and the $2n$-sample Ring-LWE problem could be weaker than the n-sample Ring-LWE problem. For large parameter sets (not proposed in practice), the NTRU problem is proven to be hard, so NTRU-based cryptosystems are based on the n-sample Ring-LWE problem.

Another related hard problem is the Ring Short Integer Solution (Ring-SIS) problem which asks if there is a short integer solution $x \in \mathbb{Z}^m$ to the equation $A \cdot x = 0 \pmod q$, for a matrix $A \in R_q^{n \times m}$.

2.5 MULTIVARIATE-SYSTEM BASED

Multivariate cryptography goes back to the late eighties and is based on the hardness of finding a solution to a system of multivariate quadratic equations over finite fields. It is possible to build signature schemes from systems of equations with uniformly random coefficients [108], and these are considered to be the most secure multivariate systems. However, the more efficient schemes use trapdoored systems of equations, which appear random to outsiders, but which have some hidden structure that is only known to the person that constructed the system. Thanks to these structures it is possible to find solutions efficiently. These are often called Oil-and-Vinegar schemes.

Currently, the multivariate encryption schemes are not very efficient, often with very large public keys and long decryption times. On the signatures front however, things look a bit better. Out of the nineteen signature schemes submitted to the NIST Post-Quantum Cryptography (PQC) project, seven were multivariate signature schemes. Two of these seven schemes proceeded to the third round of the NIST PQC process. The Rainbow scheme [41] was selected as one of the three finalists, and the GeMMS scheme [29] was selected as an "alternate candidate". These schemes enjoy very short signature sizes (as small as 33 Bytes), but suffer from rather large public keys (160 KB or more).

2.6 THE NIST ROUND 3 CANDIDATES

In the table 2.1, we describe the NIST Round 3 candidates (both the finalists and the alternate candidates) and splitting them into the two groups of encryption and signature scheme, whilst also detailing the hard problems on which they are based.

Table 2.1: NIST Round 3 candidates

Scheme	Enc/Sig	Family	Hard Problem
		Round 3 Finalists	
Classic McEliece	Enc	Code-Based	Decoding random binary Goppa codes
Crytals-Kyber	Enc	Lattice-Based	Cyclotomic Module-LWE
NTRU	Enc	Lattice-Based	Cyclotomic NTRU Problem
Saber	Enc	Lattice-Based	Cyclotomic Module-LWR
Crystals-Dilithium	Sig	Lattice-Based	Cyclotomic Module-LWE and Module-SIS
Falcon	Sig	Lattice-Based	Cyclotomic Ring-SIS
Rainbow	Sig	Multivariate-Based	Oil-and-Vinegar Trapdoor
		Round 3 Alternate Candidates	
BIKE	Enc	Code-Based	Decoding quasi-cyclic codes
HQC	Enc	Code-Based	Coding variant of Ring-LWE
Frodo-KEM	Enc	Lattice-Based	LWE
NTRU-Prime	Enc	Lattice-Based	Non-cyclotomic NTRU Problem or Ring-LWE
SIKE	Enc	Isogeny-Based	Isogeny problem with extra points
GeMSS	Sig	Multivariate-Based	'Big-Field' trapdoor
Picnic	Sig	Symmetric Crypto	Preimage resistance of a block cipher
SPHINCS+	Sig	Hash-Based	Preimage resistance of a hash function

3 SECURITY NOTIONS AND GENERIC TRANSFORMS

In this chapter we provide some general background on the objects considered in the NIST competition, the used security notions, and the most common ways to construct them. We start with a focus on PKE and KEMs. Afterwards, we move on to signatures. We conclude with a discussion of different security models considered in proofs. Naturally, our exposition remains on a rather informal level. For a more detailed, formal definition of the discussed primitives and notions see for example [70].

3.1 PKE AND KEMS

The NIST competition was initiated to find replacements for the public key primitives widely used in practice. With regard to primitives that provide authenticity this clearly means digital signature schemes. For primitives that provide secrecy, the answer is not that clear. The discussed options were as follows:

Key exchange (KEX): A protocol run between two parties. At the end of an execution, a key exchange protocol outputs the same session key at both parties (at least with high probability).

Public Key Encryption (PKE): A public key encryption scheme consists of three algorithms. The key generation algorithm generates a key pair consisting of a public and a private key. The encryption algorithm takes a message and a public key to compute a ciphertext. The decryption algorithm takes a ciphertext and a private key to compute a plaintext. We require that (at least with high probability) the decryption of an encryption returns the original message if the correct private key is used.

Key Encapsulation Mechanism (KEM): A key encapsulation mechanism consists of three algorithms, similar to PKE. Like for PKE, the key generation algorithm generates a key pair consisting of a public and a private key. In contrast to PKE, the encapsulation algorithm takes a public key to compute a session key and a ciphertext. The decapsulation algorithm takes a ciphertext and a private key to compute a session key. We require that (at least with high probability) the decapsulation of a ciphertext returns the same session key as the encapsulation that generated it if the correct private key is used.

These primitives are linked insofar as we know how to build KEMs from PKE and we know how to construct KEX from KEMs (and thereby also from PKE). After a long discussion process NIST decided that they will standardize KEMs. KEX was decided against, as all the available candidate KEX required interaction and thereby are less general than PKE and KEMs. KEMs were chosen over PKE as in most applications, the main purpose of PKE is to encrypt session keys. A KEM does essentially that, with the difference that the scheme has control over choosing the session key. This small difference makes it easier to construct secure schemes.

3.1.1 Security Notions, and Transforms

The general approach to construct public key cryptography is as follows. We start with a mathematically hard problem, like Learning-with-Errors (LWE). From this we construct a very basic primitive that usually fulfills a weak security notion, for example a weakly secure PKE scheme. We then "transform" this basic primitive into the primitive we are actually interested in, in this example a strongly-secure KEM. Security is then argued cryptanalyzing the mathematical problem and providing reductionist proofs for the constructions and transforms. These proofs relate the complexity of attacking one primitive to the complexity of attacking another primitive, or solving the mathematical problem. In the running example, we would relate the complexity of attacking the KEM to the complexity of attacking the PKE scheme, which we in turn would relate to the complexity of solving the mathematical problem. If these relations are tight, establishing that solving the mathematical problem is hard, implies that attacking the KEM is hard.

The mathematical problems are what distinguishes the different areas of post-quantum cryptography listed in the previous section. The intermediate primitives, security notions, and transforms are the same and will be discussed below.

We start with the notions and transforms used in the construction of KEMs. The security notion that we expect from a KEM is IND-CCA security, often also called active security as it models active attackers. The way to get there is via the intermediate step of constructing a weakly secure PKE scheme and then applying a generic transform. All third round candidates use a variant of the so-called Fujisaki-Okamoto (FO) [51] transform for this (for the variants see e.g. [56]). Before we outline the transform, we briefly recall the used security notions. While all three notions can be given for PKE and KEMs, we give the first two for PKE and the last for KEMs as this is how we use them later.

IND-CPA: Ciphertext-indistinguishability under chosen plaintext attacks is what is sometimes called passive security. It describes a setup in which an adversary can encrypt messages themselves (for PKE this only means the adversary knows the public key). The security notion says that given a ciphertext of an unknown message, any information about the corresponding plaintext (except its length) should be hidden from the adversary. This is modelled by allowing the adversary to choose two messages. At random one of these two messages gets encrypted and the ciphertext is given to the adversary. The adversary then has to decide which message was encrypted. IND-CPA security is only achievable by randomized PKE where the encryption algorithm is randomized (i.e., the same message is never encrypted to the same ciphertext).

OW-CPA: One-wayness under chosen plaintext attacks describes the same setup in which an adversary can encrypt messages themselves. However, security in this case only says that given the encryption of a random message (not chosen by the adversary) the adversary cannot recover the original message. This is clearly a weaker security notion than IND-CPA. However, this notion is achievable by deterministic PKE (dPKE) where the encryption of a message under a fixed public key always results in the same ciphertext.

IND-CCA: Key-indistinguishability under chosen ciphertext attacks is what is sometimes called active security. It describes a setup in which an adversary can not only generate encapsulations themselves but also has a way to learn the decapsulation of ciphertexts of their choosing. The security is then defined as follows. Given a ciphertext, any information about the key it encapsulates (except that it is a session key of a certain length) should be hidden from the adversary. This is modelled by giving the adversary a ciphertext and a session key. The session key is either the session key returned by the encapsulation when the ciphertext was generated or a randomly sampled session key that is independent of the ciphertext. The adversary then has to decide which was the case.

Note that IND-CCA security is necessary whenever a keypair is used for more than a single encapsulation. The reason is that the adversary might be able to send a victim a manipulated ciphertext. While it is usually not the case that the victim will leak the decapsulated key afterwards, the adversary might still gain information about this key from the behavior of the victim after decapsulation (e.g., in case of TLS if the victim continues to establish a session or if it aborts and how long that takes). This is not covered by IND-CPA security and indeed, we know attacks against several schemes that were submitted to the NIST competition based on finding valid ciphertexts for which decapsulation fails.

3.1.2 FO-transform

We now have everything to outline the FO-transform. Most proposals in the NIST competition start off with constructing an IND-CPA secure PKE scheme. The mathematical problem used and the way this is done is exactly where the proposals differ.

Step 1: IND-CPA PKE to OW-CPA dPKE. The first step of the FO-transform is then to de-randomize the PKE scheme. For this we note that a randomized algorithm is simply a deterministic algorithm that additionally takes randomness in form of a bit string as input. De-randomization is done by replacing the randomness input by the hash of the message. It can be shown that the result is a OW-CPA secure dPKE scheme.

Step 1: OW-CPA dPKE to IND-CCA KEM. The reason to de-randomize the PKE scheme is that this allows to add a re-encryption check in the decryption routine. I.e., given a ciphertext (and the right private key) the receiver can decrypt the ciphertext, then encrypt the result again and see if the same ciphertext is obtained. It can be shown that this has as a consequence that any ciphertext that is not chosen by encrypting a message (and without the knowledge of the private key) will be rejected because it fails the re-encryption check. The dPKE scheme is then transformed into a KEM as follows. Key generation remains the same. The encapsulation algorithm chooses a random message to be encrypted with the dPKE scheme. The ciphertext under the dPKE scheme is the KEM ciphertext. The session key is the hash of the concatenation of the message and ciphertext. The decapsulation algorithm decrypts the ciphertext and runs the re-encryption check. If the check fails, the algorithm returns a random session key. If the check succeeds, the algorithm returns the hash of the decrypted message and the ciphertext as session key.

Variations. There exist several variations. Some proposals directly construct an OW-CPA dPKE scheme and skip step 1. Beyond that, step 2 is slightly modified in some proposals. Some schemes allow to verify a ciphertext without doing a full re-encryption which can improve speed. Some proposals include the public key (or a hash thereof) into the hash function call that derives the session key. This allows to increase the cost of multi-user attacks where an adversary attacks multiple users at once. Another variation is explicit rejection where the decapsulation aborts with a clear failure code instead of terminating normally and returning a random key. This option is not widely used as it complicates the security proofs. Finally, some schemes add a so-called key confirmation hash. This is another hash of the message that gets encrypted (and possibly the PKE ciphertext) under a different hash function than the one used to compute the session key. This additional hash is then sent as part of the KEM ciphertext. This allows to replace the re-encryption by a hash function call at the cost of increased KEM ciphertext size.

3.2 DIGITAL SIGNATURES

As explained above, the situation for signatures is a bit more straight-forward than that for KEMs / PKE. There were no alternative notions to consider. A digital signature scheme is defined as follows.

Digital Signature Scheme (DSS): A DSS consists of three algorithms. The key generation algorithm generates a key pair consisting of a public and a private key. The signature algorithm takes a message and a private key and returns a signature. The verification algorithm takes a message, a signature, and a public key and returns true or false. For correctness we expect that a correctly generated signature verifies under the right public key (i.e., verification returns true).

The common notion of security for DSSs is the following notion which may be considered security against active attackers.

EUF-CMA: Existential Unforgeability under Chosen Message Attacks states that it is hard for an adversary to forge a signature on any message that was not signed by the holder of the secret key. This is modeled by a game in which the adversary is allowed to ask a signer for signatures on arbitrary messages. The adversaries goal is come up with a valid signature on a fresh message, i.e., a message that the signer was not asked to sign.

As for KEMs, we do not construct EUF-CMA-secure DSSs from scratch. Instead, we construct an intermediate building block and then use a generic transform. While for KEMs there is essentially one transform used by all proposals, this is different for DSSs. The different proposals can be split into at least three approaches[1]. To not turn this section into a book, we stick to outlines and simplify things where appropriate. We would like to refer the interested reader to the literature for the precise definitions used for the different NIST proposals.

3.2.1 Identification schemes

The most common approach in the competition is to construct an identification scheme as intermediate building block. This approach is know from (EC)DSA for example.

Identification Scheme (IDS): An identification scheme is a protocol between a prover and verifier. An IDS implicitly contains a key generation step[2] in which the prover generates a keypair consisting of a public and a private key. In the IDS protocol a prover then tries to convince a verifier that it knows the private key to a given public key. The relevant IDS protocols in this context start with the prover sending a commitment message followed by a challenge (possibly a sequence of challenges) by the verifier that is then answered by the prover. In the end, the verfier takes the public key and the full transcript of exchanged messages and runs a verification algorithm.

The security notions required from an IDS such that we can transform it into an EUF-CMA DSS are twofold.

[1] Considering more details, further distinction is possible but that goes beyond the scope of this report.

[2] IDS are often described with respect to a witness relation due to historical reasons. In that case the instance generation of sampling a statement with a witness can be considered the key generation.

HVZK: Honest-Verifier Zero-Knowledge says that a verifier that correctly follows the protocol cannot learn anything about the used secret key. This is modelled by requesting that we can build an algorithm that without knowledge of the secret key can generate valid transcripts of protocol runs that are indistinguishable from real transcripts.

Soundness: Soundness states that a verifier catches a prover that does not know the secret key with at least constant probability. The precise definition of this property depends on the kind of IDS that is used. For the easiest case of a single commit message followed by one binary challenge and response this is defined as the ability to compute the secret key from two protocol transcripts that contain the same commit message but a different challenge. This demonstrates that an adversary that can respond to both challenges for a given commitment must know the secret key.

Note that the probability to catch a cheating prover can be amplified to be arbitrarily close to one by running the scheme multiple times.

FS-transform. The common way to turn an IDS into a signature scheme is the Fiat-Shamir (FS) transform [49]. It uses a cryptographic hash function to replace the verifier in computing the challenges[3]. More precisely, the challenges are computed as hash of the message to be signed, the public key, and all previous protocol messages. The protocol transcript is then sent as signature. The verification algorithm consists of the final verification algorithm executed by the verifier in the IDS protocol and verifying that the challenges are correctly computed from the message, public key, and transcript data. For this to be secure, the IDS must have soundness that catches a cheating prover with overwhelming probability. Hence, the transformation has to be applied to the IDS obtained by running the original IDS a sufficient number of times.

3.2.2 Trapdoor Functions

The second approach, which was also used for RSA signatures, is the use of a trapdoor function. While RSA provides us with a trapdoor permutation, in the post-quantum setting the notion of trapdoor functions was generalized as follows [52]:

(Preimage-Sampleable) Trapdoor Function (TDF): A trapdoor function is a triple of algorithms. The key generation algorithm generates a key pair consisting of a public and a private key. The evaluation algorithm takes the public key (which in this case works as a function description) and a domain element and outputs an image of that element. The preimage sampling algorithm takes the private key and an element in the image, and computes a matching preimage, i.e., a domain element that is mapped to this image under the function described by the public key.

The basic security notion that a TDF has to fulfill to be suitable for the use in a signature scheme is one-wayness. In addition, the preimages generated using the private key must not leak information about the key.

One-wayness with private key: A TDF is one-way if given a random element from its image, it is hard to find a matching preimage i.e., a domain element that the evaluation algorithm maps to this image given the public key.

Indistinguishable Preimage-Sampling: A TDF has indistinguishable preimage-sampling if the distribution generated by the preimage sampling algorithm

[3] For this to work, the IDS has to be "public coin", i.e., the challenges must be chosen using the uniform distribution over some set.

using the secret key is indistinguishable from an (efficiently sampleable) distribution on the preimage space that is independent of the private key[4].

For RSA we are in the special case that we even have a trapdoor permutation. In that case the preimage sampling is always indistinguishable as there is only one preimage per image.

FDH-transform. The transformation that is used for this case is called Full-Domain Hash (FDH) and is the same as for RSA signatures [5] . The transform uses a cryptographic hash function that maps arbitrary-length messages to the domain of the TDF. For security it is important that the whole domain is in the image of the hash function (hence the name). To generate a signature, the signer hashes the message, treats the hash as an image of the TDF, and uses the preimage sampling algorithm to sample a preimage. The preimage becomes the signature.

3.2.3 One- and few-time signatures

The last approach is again following the generic idea of constructing a weaker primitive and then strengthening the security via the use of a generic transform or construction. In this case, the weaker security is defined in terms of the number of signatures that can safely be created using a key pair:

One-time Signature Scheme (OTS): A one-time signature scheme [74] is a digital signature scheme where a key pair may only be used to sign one message. The corresponding security notion is One-Time EUF-CMA. In this variant of EUF-CMA the adversary is only allowed to ask the signer for a signature on one message of its choice before coming up with a forgery.

Few-time Signature Scheme (FTS): A few-time signature scheme is a digital signature scheme where a key pair may only be used to sign a few messages (the precise number usually depends on the used parameters). The corresponding security notion is q-Time EUF-CMA. In this variant of EUF-CMA the adversary is only allowed to ask the signer for signatures on q messages of its choice before coming up with a forgery.

Actually, every signature scheme is a few-time signature scheme. However, for a full-fledged signature scheme NIST for example requires the number of signatures q to be 2^{64}. For an FTS the number of signatures is usually clearly below 100.

Merkle Signature Scheme (MSS). The generic transform to turn a one-time signature scheme into a stateful signature scheme is using Merkle trees [90]. The idea is to use a huge number of key pairs and authenticate the public keys of these key pairs using a binary hash tree called a Merkle tree. The limiting property of the resulting signature scheme is that it is stateful. The signer has to remember which OTS key pairs it already used, in order not to use one twice.

SPHINCS. The SPHINCS construction [15] turns an MSS into a stateless, i.e., general purpose, signature scheme via the use of an FTS. The fundamental idea is that an MSS is used to authenticate FTS public keys. Every OTS key pair (which are the leaves of the Merkle tree) is used to sign an FTS public key. For every message a random FTS public key is picked to sign. The SPHINCS construction is described in a bit more detail on the page for SPHINCS$^+$ in Chapter 5.

[4] Mathematically, it is required that there exists an efficiently sampleable distribution D on the domain such that the distribution of using the preimage sampling algorithm with a secret key for some image y is indistinguishable from the distribution obtained by sampling from D and accepting if the resulting value is a preimage of y.

[5] In practice we use the probabilistic version PFDH, which is used in RSA-PSS, which randomizes the hash in a certain way.

3.3 QUANTUM ADVERSARIES

In the post-quantum setting we are concerned with a world where an adversary has a quantum computer at its disposition while honest users still work on conventional computers. For the security notions above, this means that we have to assume that the adversary is a quantum algorithm. While this of course has an impact on the complexity of solving the mathematical problems used, many security proofs remain unaffected by this change. However, there are two notable exceptions. First, quantum algorithms are inherently probabilistic. This means that arguments that use techniques like rewinding, which try to run an algorithm with the same random coins twice, do not trivially translate to the post-quantum setting. The second exception is the random oracle model discussed in the following chapter.

3.3.1 The quantum-accessible random oracle model

The security proofs that are usually given for practical systems (e.g., RSA-PSS, RSA-OAEP, ECDSA, etc.) belong to a heuristic proof model called the random oracle model (ROM). In this model, the cryptographic hash function used is modeled as an ideal random function for the security proof. This random function is called a random oracle. It is modeled as an oracle that every participant in the system has access to and that behaves like a random function; specifically its answers are consistent, and answers to new queries are uniformly distributed. While there exist artificial counterexamples of schemes that are secure in the ROM, but cannot be secure for any cryptographic hash function, the ROM has proven to be a reliable tool for the analysis of real world systems.

In the post-quantum setting we consider adversaries to have access to a quantum computer. As hash functions are public functions, this means that an adversary can run these on its quantum computer and, consequently, can query these on superpositions of inputs or even on entangled quantum states. This ability cannot be reflected in the conventional ROM. Hence, the QROM (quantum-accessible ROM) was proposed [22]. In the QROM, adversaries are given exactly this additional ability. By now we know separations between the ROM and the QROM, i.e., cryptographic schemes that can be proven secure in the ROM but that are inherently insecure in the QROM. This demonstrates that a security proof in the ROM is only of limited value when aiming for post-quantum security.

For most of the transforms above, proofs are only known in the ROM and or QROM (cf., [19, 43, 64, 107]) but not in the standard model (i.e., without making idealizing, heuristic assumptions).

4 NIST ROUND 3 FINALISTS

4.1 ENCRYPTION SCHEMES

4.1.1 Classic McEliece

Design:

Classic McEliece [3] is a code-based scheme using binary Goppa codes, the same codes that McEliece originally proposed when he introduced code-based cryptography [86] in 1978. Code-based cryptography is the oldest public-key encryption system that is expected to resist attacks by quantum computers and is one of the oldest public-key encryption systems overall. During Round 2 the scheme merged with NTS-KEM, which was using the same codes.

The assumption underlying One-Wayness against Chosen-Plaintext Attacks (OW-CPA) PKE security is that decoding a random binary Goppa code is hard – McEliece encodes messages into code words and encrypts them by adding random errors. The Classic McEliece scheme uses the dual of McEliece's scheme, as proposed by Niederreiter [93], and tightly turns this OW-CPA PKE into an IND-CCA2 KEM using a variant of the FO-transform (cf. Sect. 3.1.2), specifically Theorem 8 in Dent [40]. A proof in the QROM (Quantum Random-Oracle Model) is given in [19] which proves a bound ϵ on the probability of a QROM IND-CCA2 attack, assuming a bound on the scale of ϵ^2 on the probability of an OW-CPA attack against the underlying deterministic PKE.

Implementation:

A full KEM was specified and implemented in [13] with improvements in [31]. The software is available on the submitters' page, see [3], and includes reference and optimized implementation. All implementations of Classic McEliece are constant time. An implementation for the ARM Cortex-M4 is finished, but not yet publicly available. FPGA implementations are covered in [115] and [116] and are also freely available and constant time.

Classic McEliece has been integrated into the network protocols McTiny [17] and Post-quantum WireGuard [61].

Cryptanalysis:

There are two main avenues of attack against code-based cryptography: information-set decoding (ISD) and structural attacks.

ISD goes back to a general decoding technique from 1962 due to Prange [102]. There is a long history of research on this problem, especially for cryptographic applications, with the most recent papers being [25, 26, 71]. These attacks show their biggest effect for high-rate codes while the binary Goppa codes used in Classic McEliece are only marginally affected. More precisely, achieving 2^λ security against Prange's attack requires keys of size $(0.741186\ldots + o(1))\lambda^2(\log_2 \lambda)^2$ bits as $\lambda \to \infty$. To achieve the same level of security against all the later attacks requires keys of size $(0.741186\ldots + o(1))\lambda^2(\log_2 \lambda)^2$ bits as $\lambda \to \infty$, i.e., the improvements affect only the $o(1)$ term. All these attacks involve huge searches, like attacking AES. The quantum attacks (Grover etc.) leave at least half of the bits of security.

Structural attacks attempt to find a good decoding algorithm for the code in the public key by identifying structures of the private key in the public one. Such attacks have been successful against code-based systems based on other codes, e.g., identifying Reed-Solomon codes as used by Niederreiter [93] or Gabidulin codes used in early rank-metric codes. However, for binary Goppa codes the only attacks known are distinguishing attacks and even those are successful only for very high-rate codes, larger than proposed for any cryptosystems [48].

Advantages and Disadvantages:

The advantages for Classic McEliece are that it has a very long history of analysis with no significant impact on the security and that the ciphertext size is small. The ciphertexts are the smallest of all Round-2 candidates and thus also of all Round-3 candidates. No other public-key encryption system can look back at more than 40 years of cryptanalysis – quantum or not – without taking a hit.

The disadvantage is the size of the public key, which for the highest security level takes more than 1MB. This poses a problem for applications that request fresh public keys for each execution; the McTiny protocol [17] shows how to make this work nevertheless without causing denial-of-service attack on the servers. Post-quantum WireGuard [61] and PGP are applications where the system can be used as a long-term identity key.

4.1.2 Crystals-Kyber

Design:

Kyber is an Indistinguishability under Chosen Plaintext Attack (IND-CCA) secure KEM originally presented in [23]. It has seen some significant changes since then and the latest description can be found in [111]. The security of Kyber can be provably reduced to the Module-Learning-with-Errors problem (Module-LWE), but the parameter set for the lowest security level bases its security estimate on a combination of Module Learning with Errors and Module Learning with Rounding (MLWR). Kyber is based on LPR [81] encryption, but uses vectors of polynomials as elements, performs additional compression on the ciphertext and is designed to accommodate fast multiplications using the Number Theoretic Transform (NTT). IND-CCA security is obtained through a variant of the FO transformation (cf. Sect. 3.1.2). The public key sizes of Kyber are 800, 1184 and 1568 bytes for security levels 1, 3 and 5 respectively, and the ciphertext sizes are 768, 1088, 1568 bytes.

Implementation:

After an initial implementation on general purpose processors in [23], Kyber has been implemented on Cortex-M4 [27] and a software hardware codesign has been described in [36]. An implementation using an RSA-coprocessor was given in [5]. Moreover, implementations of Kyber can reuse existing designs for Ring-LWE (aka RLWE) encryption schemes that support NTT multiplication, for example implementations of NewHope or early Ring-LWE schemes. No side-channel secure implementation is available for Kyber, but an idea of the challenges and the cost can be gained from a masked Ring-LWE implementation as presented in [96].

Cryptanalysis:

The security of Kyber is provably reducible to the security of the underlying Module-LWE problem (aka Mod-LWE). As there is currently no efficient way to exploit the modular structure security is typically estimated based on the corresponding LWE problem. Such attack typically transforms the LWE problem into a shortest vector lattice problem that can then be solved using lattice reduction techniques. An independent security estimate of Kyber was given in [4].

Kyber has a very small probability of decryption failures in which valid ciphertexts fail to decrypt properly. This paves the road for decryption failure attacks as proposed in [21, 37, 39]. However, when limiting the number of queries to 2^{64} as recommended in the NIST call for proposals [94], these attacks are less efficient than direct lattice attacks. A practical fault injection attack on Kyber was presented in [105].

Advantages and Disadvantages:

Kyber is designed with NTT multiplications in mind, which allows for efficient implementations of Kyber on a variety of platforms. It is notable that some elements are generated and compressed in the NTT domain, which makes it impractical to use other multiplication algorithms for Kyber. Moreover, polynomial multiplications are in the same ring for all security levels, which makes it easy to scale between the security levels. Overall, the support for NTT multiplication makes Kyber efficient to implement. The security of Kyber enjoys strong reductions to underlying hard lattice problems.

4.1.3 NTRU

Design:

Nth Degree Truncated Polynomial Ring Units (NTRU) is one of the oldest encryption schemes that makes use of structured lattices. It was developed by Hoffstein, Pipher, and Silverman in 1998 [55]. The round three submission to NIST [118] is a merger of the initial NTRU submission [117] and the NTRU-HRSS submission [110] implemented after the first round due to large overlaps in the design. The submission specifies a perfectly correct, deterministic public key encryption scheme (dPKE). This dPKE is transformed into a CCA2-secure KEM using a variant of the FO-transform (cf. Sect. 3.1.2), specifically the $\mathrm{U}_m^{\not\perp}$ transform of [56]. For this transformation, a tight proof for CCA2-security in the ROM is given in [56]. A tight proof in the quantum-accessible ROM is known, but makes a less standard assumption [107].

Implementation:

The NTRU-HRSS part of the submission was based on [60] which already contained a high-speed constant-time implementation. NTRU-HRSS was among the fastest first round submissions. NTRU is also known for its speed on constrained devices; implementations go back to at least 2001 [8], but also nowadays NTRU is one of the schemes with the fastest encapsulation and decapsulation routines in the pqm4 project [67].

Also, implementation security of NTRU is well advanced. As mentioned above, for commodity hardware, the optimized implementations provided are constant time [60]. On constrained devices, up-to-date masked implementations are known [109] that protect against side channel attacks like correlation power analysis attacks [78].

NTRU was chosen by Cloudflare and Google for their second PQC experiment [77] and used in connections from users running Chrome Canary to Google and Cloudflare.

Cryptanalysis:

The security of NTRU is supported by a long history of cryptanalysis (see e.g., [33, 54, 58, 83, 84]). Up to parameter changes, NTRU successfully survived the last 20+ years of cryptanalysis. The efforts of the last years suggest that the complexity of the best attacks against NTRU is determined by the complexity of lattice reduction. The complexity of the best algorithms for lattice reduction in turn depends on the complexity of solving the shortest vector problem (SVP). See the specification for an extensive elaboration. An independent evaluation can be found in [4].

Advantages and Disadvantages:

NTRU has several advantages. As mentioned above, it is perfectly correct and the underlying assumption is well studied. It is flexible, meaning that the underlying dPKE can be parameterized for a variety of use cases with different size, security, and efficiency requirements. It is simple: The dPKE has only two parameters, n and q, and can be described entirely in terms of simple integer polynomial arithmetic. It is fast: ntruhrss701 was among the fastest submissions in the first round. It is compact: The ntruhps2048677 parameter set achieves NIST level L1 security with a wide security margin, level L3 security under a reasonable assumption, and has public keys and ciphertexts of only 930 bytes. It is guaranteed patent free as the relevant patents have expired.

On the downside, NTRU is unlikely to be the fastest, most compact, or most secure submission. However, it is competitive on products of these measures. As for all other lattice-based schemes, the choice of optimal parameters for NTRU is currently limited by a poor understanding of the non-asymptotic behaviour of new algorithms for SVP. Finally, there is structure in NTRU that is not strictly necessary, and this may also be seen as a limitation.

4.1.4 Saber

Design:

Saber is a family of cryptographic primitives that includes an IND-CPA secure encryption scheme and an IND-CCA secure KEM, with an initial design as described in [38] and most recent update in [10]. Its security can be reduced to the security of the Module Learning with Rounding (MLWR). As most LWE/LWR based schemes, Saber follows the general structure of LPR [81] encryption. The main differences are power-of-two moduli, the use of vectors of polynomials and the adaptation of learning with rounding. To achieve IND-CCA security Saber relies on a post-quantum variant of the FO-transformatin (cf. Sect. 3.1.2). Saber boasts public key sizes of 672, 992 and 1312 bytes; and ciphertext sizes of 736, 1088, 1472 bytes for security level 1, 3 and 5 respectively.

Implementation:

An initial implementation of Saber on high end processors was presented in [38]. Implementation efforts have since then extended to Cortex-M4 and Cortex-M0 in [66, 68, 89, 98], ESP32 in [114], specific coprocessors in [82, 106], large integer coprocessors in [24], a software hardware codesign in [36] and a hardware implementation in [119]. An implementation that batches multiple decapsulations to exploit vector instructions has been proposed in [112]. A first order masked implementation of Saber was given in [11].

Saber has been integrated into the network protocol Post-quantum WireGuard [61] for exchanging ephemeral keys.

Cryptanalysis:

The most straightforward attack on Saber is to break the underlying Mod-LWR problem. Such an attack rewrites the Mod-LWR problem as a shortest vector lattice problem and uses lattice reduction algorithms to retrieve the secret key. The security of this problem is typically estimated as the security of the analogous LWE problem as there is at the moment no efficient attack that exploits the module or rounding structure. An initial security estimate of Saber was given in [4] and was further improved in [10] using the estimation tools of [2, 35].

As Saber is subject to decryption failures with a small probability, there is the possibility of decryption failure attacks. Attacks on the IND-CCA secured KEM were presented in [21, 37, 39] but when limiting the number of queries that can be performed to 2^{64} as proposed in the NIST call for proposals [94], these attacks do not outperform standard lattice attacks.

Advantages and Disadvantages:

The choice for power-of-two moduli avoids the need for explicit modular reductions or rejection sampling that are typically present in prime moduli based schemes. The latter also reduces the number of hash function calls. The drawback of this choice is that the NTT is not naturally supported. However, other multiplication algorithms (e.g., Karatsuba, Toom-Cook, schoolbook, Kronecker) have been shown to be efficient on a range of platforms and the design of Saber does not restrict implementors to a specific multiplication algorithm. Moreover, in multiplications of Saber, one element will always have small coefficients, which could be exploited for optimizing implementations.

Being based on learning with rounding, Saber introduced an error by rounding coefficients. This naturally reduces the communication bandwidth and avoids the generation of the error term. The modular structure of Saber implies that multiplications of polynomials are always in the same ring, and as such the multiplication algorithm of these polynomials is the same for all security levels.

Saber is efficient to mask, due to the power-of-two moduli and the absence of the error term. The first order masked Saber implementation of [11] has an overhead factor 2.5x, which can be compared to an overhead of factor 5.7x previously reported for prime-moduli schemes [96]. Saber also excels at anonymous communication as it is naturally constant time, even over different public keys, due to the avoidance of rejection sampling. Moreover, the power-of-two moduli ensures communication consists of a uniformly random bitstring without apparent structure.

4.2 SIGNATURE SCHEMES

4.2.1 Crystals-Dilithium

Design:

Dilithium is a signature scheme introduced in [45] and with latest version described in [80]. It follows the concept of designing signature schemes from identification schemes (cf. Sect. 3.2.1), using Fiat-Shamir with aborts.Its security can be reduced to the security of the Module-Learning With Errors (MLWE) and Module Short Integer Solution (MSIS) problems. It is designed to allow fast multiplications using the NTT transformation and avoids generation of randomness from a discrete Gaussian distribution, instead opting for sampling from a uniform distribution.

Implementation:

The Dilithium team provided an implementation in their initial work [45]. Further work has focused on improving the speed of the signing procedure [104]. An implementation of Dilithium on Cortex-M4 was presented in [53] and a masked implementation was introduced in [91].

Cryptanalysis:

The security of Dilithium is based on that of the underlying MLWE and MSIS problems. Currently there is no efficient attack exploiting the module structure and as such the security of the equivalent LWE and SIS problems is considered. An independent estimation effort [4] confirmed Dilithium's security estimate. A fault attack on Dilithium was presented in [28].

Advantages and Disadvantages:

In contrast to other signature proposals, Dilitihium samples from a uniform distribution avoiding the complex and inefficient sampling from a discrete Gaussian distribution. The modular structure of Dilithium ensures that polynomial multiplication is always performed in the same ring regardless of security level, which makes it easy to switch between these levels. Multiplication can be performed efficiently due to its NTT friendly parameters. Applying a trick to compress the public key with a factor 2, Dilithium has the smallest public key plus signature size of lattice-based schemes that use uniform sampling.

4.2.2 Falcon

Design:

Falcon [103] is a signature scheme whose design is based on the Gentry–Peikert–Vaikuntanathan (GPV) blueprint [52] for lattice-based signatures using (preimage-sampleable) trapdoor functions (cf. Sect 3.2.2). It instantiates this construction with NTRU lattices and an efficient Gaussian sampler [46,57], which yields a scheme that is provably secure under the assumption that SIS is hard in the particular lattices used. Falcon has been designed so that all of the arithmetic operations can be computed using efficient Fourier-transform techniques.

Implementation:

An efficient constant-time implementation of Falcon is given by [101], using the sampler of [57]. It does not require (but can use) a floating-point unit and runs efficiently on various kinds of microprocessors including Intel x86 and ARM cores. See [97] for a more optimized implementation specific to the latter. The constant-time Gaussian sampler of [69] can be used in Falcon.

Cryptanalysis:

The mathematical security of Falcon relies on the hardness of the SIS problem over NTRU rings, which benefits from the long history of cryptanalysis for the NTRU cryptosystem (cf. Section 4.1.3). The best known attacks are generic lattice techniques: there is no known way to effectively exploit the additional ring structure present in NTRU lattices. To estimate the security against lattice-reduction algorithms, Falcon employs the "Core-SVP" method which was also used by many other lattice-based NIST submissions.

A fault attack on Falcon is demonstrated (and countermeasures proposed) in [85], and the side-channel leakage of Falcon and similar schemes was analysed in [50].

Advantages and Disadvantages:

In a nutshell, Falcon is a very compact (smallest combined size of public key and signature among all NIST candidates) and efficient post-quantum signature scheme whose security reduces to well-established assumptions. The chosen ring structure and error distribution allow for efficient FFT-based implementations, which partially cancels the adverse effects of using a Gaussian error distribution and leads to good performance in practice. Indeed, perhaps the biggest drawback of Falcon appears to be the complexity of understanding all details of the construction and implementing the scheme correctly.

4.2.3 Rainbow

Design:

Rainbow is a multivariate signature scheme, proposed by Ding and Schmidt [41, 42] and based on the Oil and Vinegar (OV) scheme by Patarin [99]. Rainbow follows the idea to construct signatures from trapdoor functions (cf. Sect. 3.2.2). Similar to RSA signatures, Rainbow uses a trapdoor function $\mathcal{P}$, for which only the holder of the secret key can compute preimages. To sign a message M, the signer then publishes a preimage for $\mathcal{H}(M, \mathsf{salt})$, where $\mathcal{H}$ is a cryptographic hash function that outputs elements in the range of $\mathcal{P}$, and where salt is a fixed-length bitstring, chosen uniformly at random for each signature.

The Rainbow trapdoor function is best described as the composition of two or more oil and vinegar trapdoors. The design philosophy is that by iterating the OV trapdoor, it gets more resistant to attacks, which allows for more efficient parameter choices. Unfortunately, the additional complexity also opens up some new attack strategies.

Implementation:

The Rainbow team provided an optimized implementation for general purpose processors and for processors supporting AVX2 instructions. These implementations are claimed to resist timing side-channel attacks. During the second round of the NIST PQC process, the Rainbow team switched to a new key generation algorithm. This does not affect the security of the scheme, but made key-generation more efficient. A fault attack against Rainbow is presented in [73].

Cryptanalysis:

Like most multivariate signature schemes, Rainbow does not have a security proof that reduces a hard computational problem to the security of the scheme. Therefore, we can not rely on widely believed assumptions and it necessary to have a dedicated cryptanalysis of Rainbow. After some initial cryptanalytic results in the first few years after the introduction of Rainbow, the cryptanalysis of Rainbow was relatively stable. However, since Rainbow entered the NIST PQC process, there have been some works that slightly improved existing attacks [9, 113], and during the third round of the NIST PQC process two new attacks were published that broke the security claims. [18] The Rainbow team has announced that a new parameter set will be proposed to address the new attacks.

Advantages and Disadvantages:

Rainbow signatures are small (e.g. ~ 66 Bytes at SL I) and the signing and verification algorithms are fast. Rainbow uses only linear algebra over very small finite fields, which makes it suitable for implementing the scheme on low-cost devices, without the need for a cryptographic coprocessor. On the other hand, the public keys are rather large (e.g. 158 KB at SL I). It is possible to compress the public key size by almost a factor 3 at the expense of slower signing times. The security analysis of Rainbow cannot be considered stable at the moment.

5 ALTERNATE CANDIDATES

5.1 ENCRYPTION SCHEMES

BIKE

BIKE [7], Bit Flipping Key Encapsulation, is a Key Encapsulation Mechanism (KEM) based on quasi-cyclic codes with moderate-density parity-check matrices. The public key specifies an error-correcting code, as in Classic McEliece, but in BIKE the code has a public structure of being quasi-cyclic, allowing the public key to be compressed. The moderate-density parity-check matrices are secret, Bit flipping corrects errors by repeatedly flipping the input bits that, given the secret parity checks, seem most likely to be errors.

HQC

HQC [88], Hamming Quasi-Cyclic, has the same noisy Diffie–Hellman structure as many lattice-based cryptosystems. The public key includes a random G and $A = aG + e$, where a, e are small secrets. The ciphertext includes $B = bG + d$ and $C = M + bA + c$, where b, c, d are small secrets and M is a message encoded using an error-correcting code. The receiver computes $C - aB = M + be + c - ad$, which is close to M since a, b, c, d, e are small, and decodes the error-correcting code to recover M. HQC uses polynomials modulo 2, rather than the larger integer moduli used in lattice-based cryptosystems, but uses polynomial modulus $x^n - 1$ with relatively large n. HQC uses error-correcting codes built from Reed-Muller and Reed-Solomon codes. Public keys are between 2249 and 7245 bytes, and ciphertexts are between 4481 and 14469 bytes, depending on the security level.

Frodo-KEM

FrodoKEM [92] is a key encapsulation mechanism whose security is based on the hardness of the standard Learning With Errors problem. The algorithm is a specific instantiation of the construction of Lindner and Peikert from 2011 [79]. It thus makes no use of so-called structured lattices (such as those based on Ring or Module LWE), this means that the performance is not as good as the lattice based schemes selected to be the main candidates in Round 3. However, for those worried about the structural properties of these latter candidates, Frodo-KEM may be an option.

NTRU-Prime

NTRU Prime [12, 14] is a lattice-based key encapsulation mechanism (KEM) with two options: Streamlined NTRU Prime, which is similar to NTRU, and NTRU LPRime, which is similar to Kyber and SABER. NTRU Prime uses a polynomial $x^p - x - 1$ with a maximum-size Galois group (superexponential in the degree) while NTRU, Kyber, and SABER use cyclotomic polynomials with a minimum-size Galois group (linear in the degree). The original STOC 2009 Gentry FHE system and the original multilinear-map system are broken for cyclotomics but not for $x^p - x - 1$; NTRU Prime predates these attacks and is designed to protect lattice-based cryptosystems against the possibility of cyclotomic attacks. Compared to the performance of NTRU, Kyber, and SABER, the performance of NTRU Prime is sometimes slightly worse and sometimes slightly better, but is generally similar.

SIKE

SIKE [63] is a key encapsulation mechanism based on the hard problem of pseudo-random walks in supersingular isogeny graphs. This is a relatively new problem in the cryptographic arena, but the problem of studying isogenies of supersingular elliptic curves is an old mathematical problem. The main advantage of isogeny based schemes is their small public key and ciphertext size. The key problems associated with SIKE is that the performance is currently not competitive with the other proposals. This may improve however over time.

5.2 SIGNATURE SCHEMES

GeMSS

The GeMMS scheme [29] builds on a line of work that goes back to 1988; schemes in this line of work are called “Big Field” schemes. The public key for GeMMS is a multivariate quadratic system of equations over $\mathbb{F}_2$. The main idea behind “Big Field” schemes is that there is a secret change of variables that turns the public key into a (perturbed version of) a system that models a low-degree univariate polynomial equation over an extension field $\mathbb{F}_{2^n}$. Since it is possible to efficiently find the solutions to a low degree univariate polynomial, this allows someone who knows the secret change of variables to sign messages. The size of GeMMS signatures is exceptionally small, with a size of only 258 bits at NIST security level I. The main drawbacks, however, are that, with 350KB, the public keys are large, and that signing is slow, especially for the more conservative parameter choices.

Picnic

The Picnic signature scheme [1] , currently on its third iteration [65], is unique among the other candidates due to its use of the “MPC-in-the-head” paradigm [62]. In this framework, a proving algorithm simulates a virtual MPC protocol which computes the circuit for an NP relation R, e.g. $x \sim_R y \iff y = \texttt{SHA-256}(x)$. By revealing the views of a random subset of the MPC parties, this forms an interactive zero-knowledge proof of knowledge (ZKPoK) of a witness for R. In Picnic, this ZKPoK is made non-interactive and turned into a signature scheme using the traditional Fiat-Shamir transform (cf. Sect. 3.2.1); furthermore, the design uses the `LowMC` block cipher for the relation R due to this cipher’s explicit design for efficient computation in MPC [2] After several iterations in the design, the current specification document for `Picnic3` lists signature sizes of 12.6kB, 27.5kB and 48.7kB for the L1, L3 and L5 NIST security levels, respectively [65].

SPHINCS$^+$

SPHINCS$^+$ [16] is a framework that describes a family of hash-based signature schemes [3] . Using an arbitrary, secure cryptographic hash function, a signature scheme can be obtained using the SPHINCS$^+$ framework, a variant of the SPHINCS construction (cf. Sect. 3.2.3). This is in contrast to all other signature schemes

[1] See https://microsoft.github.io/Picnic/ for the project page and a list of design and specification documents. Last accessed May 2021.

[2] While producing efficient and short signatures, the use of the new `LowMC` has been commented on by NIST and other works have explored using more trusted ciphers as replacement.

[3] See https://sphincs.org for the project page with the full submission package and a collection of relevant design documents. Last accessed May 2021.

mentioned in this document [4], which require a secure cryptographic hash function and an additional mathematical problem to be computationally hard to solve. The general concept of building signature schemes from cryptographic hash functions goes back to the beginning of public key cryptography [74, 90]. For that reason, SPHINCS$^+$ is widely considered the signature scheme with the most conservative security guarantees in the competition.

The rough concept of SPHINCS$^+$ (as well as its predecessor SPHINCS and the first round scheme Gravity-SPHINCS) is as follows. A key pair defines a huge virtual data structure. Data objects required in a signature operation are generated on the fly from a short secret seed using a pseudorandom generator. This virtual data structure of a key pair contains a massive number of hash-based few-time signature scheme (FTS) key pairs (e.g. 2^{60}). Such FTS become less secure with every signature and after a certain number T of signatures (e.g. $T = 8$) security drops below the targeted security level. To prevent using the same few-time key pair more than T times, for every signature a random FTS key pair is selected for every new message. By using sufficiently many FTS key pairs, the probability of a $T + 1$ times collision can be made about as likely as successfully guessing the secret key. The public keys of all these FTS key pairs are authenticated by a single hash value using certification trees (similar to a PKI) built of hash-based one-time signature schemes and binary hash trees.

The SPHINCS$^+$ submission to the NIST process defines instances using SHA2, SHA3, or Haraka [72]. The SPHINCS$^+$ design remained unchanged since the initial submission. The changes introduced in the last iterations were an additional construction for the internally used functions and parameters that offer better performance trade-offs. SPHINCS$^+$ is a flexible design. For example, at NIST security level L1, the specification contains parameters that lead signature sizes of 7 856 bytes and 17 088, while signing times are 2 721 Mcycles and 138 Mcycles, respectively, using SHA2-256. Verification speed is generally fast with about 3 and 8 Mcycles for above parameters, and keys for both parameter sets are 64 bytes for the secret and 32 bytes for the public keys.

[4] While this is theoretically also true for Picnic, to be competitive, Picnic requires a function with low multiplicative depth, a property common hash functions do not provide.

6 QUANTUM MITIGATION

If you encrypt data that needs to be kept confidential for more than 10 years and an attacker could gain access to the ciphertext you need to take action now to protect your data. Otherwise, security will be compromised as soon as the attacker also gets access to a large quantum computer. Given that the NIST process will still run for a few years, there are essentially two viable options to handle this problem.

The first option is to already migrate to so called hybrid implementations that use a combination of pre-quantum and post-quantum schemes. The second option is to employ the conceptionally easy, but organizationally complicated measure of mixing pre-shared keys into all keys established via public-key cryptography. We will detail these two options below.

If you build devices that will be hard to reach or to upgrade later you should include a post-quantum signature scheme now to ensure secure continuity of service when a quantum computer is available. Otherwise, you should start to prepare for migration by making a catalogue of where you currently use public-key cryptography and for what purpose. Make sure to include software updates and third party products in your overview. Figure out whether you fit into one of the use cases that NIST considers – even better, get involved in the NIST discussions to make sure your use case is covered. Then wait for the outcome of the NIST competition (or quantum computers getting dangerously close, whichever comes first) to update your systems.

6.1 HYBRID SCHEMES

A hybrid scheme in this context describes the combination of a pre-quantum public key cryptographic scheme, such as RSA or (EC)DH, with a post-quantum one in a way that guarantees security as long as at least one of the two schemes is secure. Hence, hybrid solutions might also be interesting for the migration to standardized post-quantum schemes as they can be easier justified in cases where certification and compliance are an issue.

We first look at public-key encryption (PKE) and key exchange (KEX). The most generic way to combine two PKE or KEX schemes is to run both schemes to obtain one shared secret per scheme and to xor the two shared secrets to obtain a combined one. For protocols that derive a session key by means of feeding a premaster secret, obtained via public-key cryptography, into a key derivation function (KDF), it is also possible to establish one pre-master secret per scheme and to feed the concatenation of the two pre-master secrets into the KDF. This would for example be applicable in the context of TLS. An extensive case-study of combining schemes for confidentiality that takes a rather applied angle can be found in [34].

When it comes to signature schemes, the combination of two schemes is generically best handled by using them independently. This means, distributing two public keys (possibly in one certificate) and always sending two signatures, one per scheme. For specific schemes, more efficient combiners might be possible but this is a topic of ongoing research. A more detailed discussion including a discussion of practical implementations of such combiners is presented in [20].

6.2 PROTECTIVE MEASURES FOR PRE-QUANTUM CRYPTOGRAPHY

Users who do not want to embark on deploying post-quantum systems before they are standardised; yet are concerned about the long-term confidentiality of their transmitted data, can protect their systems by including retained shared secret data in the key derivation, in addition to the key material obtained by a public key operation. This comes at the expense of keeping pairwise shared data and is thus only an option for systems which keep state and have a limited set of peers.

ZRTP [120] includes such a mechanism called "key continuity" as a measure against man-in-the-middle (MITM) attacks. The protocol – specified in 2006 – does not mention security against quantum adversaries as a motivation but is the first description of this idea that we are aware of. It also goes further than other protocols in updating the shared secret data. The more recent Wireguard [44] protocol uses a pre-shared key (PSK) and includes it in the derivation of session keys but does not update the PSK; Wireguard is based on Noise PSK [100, Chapter 9]. Wireguard explicitly mentions the PSK as a feature to protect against later compromise by quantum attackers. (See also [6] for a small tweak to achieve better protection in that scenario and [61] for a fully post-quantum version.)

The following description follows the approach of ZRTP in that the retained shared secret gets updated with each public-key operation by hashing in new data. Including secret data from public-key operations ensures forward secrecy and post-compromise security against pre-quantum attackers. Updating the retained shared secret during each iteration with a hash function ensures that a later compromise of the system cannot recover previous session keys from the retained shared secret and recorded connection data, even if the attacker has a quantum computer and can thus break the pre-quantum public-key encryption.

Let r denote the retained shared secret. Let s be the fresh shared data, obtained from a public-key operation. The above-mentioned protocols are based on the Diffie-Hellman key exchange, but this approach can also be used for RSA-based protocols. Whenever the original protocol calls a KDF for generating the session key k, the KDF's inputs should be extended to include r:

$$k = \mathrm{KDF}(s, ''\texttt{session key}'', r, *),$$

where $*$ is a placeholder for the context data (handshake messages, public keys, ID strings, etc.). This ensure that an attacker can recover k only if he has obtained r as well as s.

After computing k, the retained secret should be updated to

$$r' = \mathrm{KDF}(k, ''\texttt{retained secret}'')$$

possibly including other context data in the KDF arguments.

The protocol needs to be careful to verify that both parties have obtained s before overwriting r. See ZRTP [120] for an instantiation using two variables for retained secret values in order to avoid desynchronization.

The description above leaves open how the users have received the first PSK value r. Users concerned about long-term security should arrange to share such keys out of band (scanned QR code, password, ...). In scenarios with predefined communication patterns, such as a main server communicating with remote registered devices, the PSK may be provisioned with the devices. Note that each device should get a unique PSK known only to the device and the server.

Users may also start with empty r if they achieve authenticity and protection against MITM attacks in other ways, e.g., comparing fingerprints of the obtained data through a different medium (a phone call etc.), or accept trust-on-first use. Note that this helps against quantum attackers only if the attackers miss the first connection, which is unlikely for an attacker so dedicated that they can get a quantum computer. However, it is worth mentioning that, if an attacker ever misses the communication leading to a key update, so that they do not know s, they also cannot compute later values of r. Hence the system can achieve security at a later state.

Note that the above approach is not suitable for systems that get restored from previously saved images, such as virtual machines. In that case a system with a fixed PSK is more suitable, however it does not protect against attackers that later get access to the system, and thus the PSK, and have recorded all messages exchanged, thus all public-key operations.

7 CONCLUSIONS

It is perhaps inevitable that as the technology sector advances drastically over time, our infrastructures are exposed to new attacking vectors. However, Quantum Technology and in particular Quantum Computing are set to be a major disruptor. We have known for more than 2 decades that the development of a sufficiently large and practical quantum computing machine will render most cryptographic systems insecure, radically transforming the existing threat model and endangering our infrastructure.

Moreover, while current systems do not pose any threat, a working quantum computer (i.e., one having a sufficient number of Qubits that is resistant to quantum noise and other quantum-decoherence, is economically viable and practically operational) is the objective of several ongoing large scale investments from both industry players and nation states. However, not all development in the area is public and it is fairly likely that the first fully functional large quantum computer will not be publicly announced. As such, policy maker and system owner should make preparations.

Rolling out new cryptographic systems takes a lot of time and effort; it might even be infeasible for systems with restricted accessibility, like satellites. Moreover, signatures play a significant role in protecting modern operating-system upgrades. If a post-quantum signature system is not in place by the time an attacker has access to a quantum computer, then the attacker can take control of the operating system through a fake upgrade and prevent any future upgrades from fixing the problem.

It is thus important to have replacements in place well in advance. What makes matters worse is that any encrypted communication intercepted today can be decrypted by the attacker as soon as he has access to a large quantum computer, whether in 5, 10 or 20 years from now; an attack known as retrospective decryption.

In this study we have provided a brief background of post quantum cryptography, in section 2 we present the 5 main families of quantum resistant cryptographic algorithms that are proposed as potential candidates to provide post-quantum security resilience; viz. code-based, isogeny-based, hash-based, lattice-based and multivariate-based. Section 4 presents the finalist algorithms that are competing to be considered by NIST ready for standardisation, whereas section 5 refers to the algorithms that NIST considers promising, but still not ready to be applied.

The last section – section 6 – presents and briefly explains two possible mitigation mechanics; namely the so-called *hybrid implementations* that use a combination of pre-quantum and post-quantum schemes, and the conceptionally easy, but organizationally complicated measure of *mixing pre-shared keys* into all keys established via public-key cryptography. Both methods have shortcomings, but for system owners requiring long term confidentiality of transmitted data are worth considering. Given that the NIST PQC standardisation process is scheduled to publish a draft standard somewhere in 2022-2024, system owners with more relaxed security requirements and/or with greater resource constraints might be better served waiting for the process conclusion.

The presented algorithms, on sections 4 and 5, refer to asymmetric key (public-key) cryptographic systems – the area of cryptography that will be mostly affected by the existence of quantum computers due to their high reliance on mathematical structures (e.g., factoring, and discrete logarithm problem). Symmetric key (shared-key) cryptographic systems on the other hand present a higher resilience to the new status-quo. In such systems, the adoption of larger key-sizes is considered an effective mitigation technique that is easy to be adopted.

The apt reader will have noticed the absence of mention of Quantum Key Distribution (QKD) [1] or of Quantum Cryptography in this text. This has been a deliberate choice. QKD is a quantum technology application that has been available for many years. It provides a guaranteed, by the laws of physics, secure way of distributing and sharing secret keys that are necessary for cryptographic protocols. It essentially offers key agreement services, but not authentication or message confidentiality; for these services we need to rely on math-based cryptography. In other words, QKD can complement a traditional cryptographic system and its setup relies on pre-established authenticated communications channels. However, the existence of such an authenticated channel, presupposes that communicating parties either have managed to privately exchanged a symmetric key in the past (e.g., by physically meeting) or are using public key cryptography. In the former case, authentication was achieved by direct interaction, which is not a scalable practice. While, in the latter, we are forced to use the same cryptographic algorithms that, as we established, are insecure against quantum cryptanalysis. It is clear that QKD is not a direct solution to the problems of quantum cryptanalysis, but rather a comparatively mature application of quantum technology. The term Quantum Cryptography, on the other hand, is often used to denote QKD or erroneously to signify Post-Quantum algorithms like the ones visited in this report. Nevertheless, it can also refer to more exotic cryptographic applications that exploit quantum properties; like quantum [pseudo]random number generators (QRNG), program obfuscation etc. It is important to note that being a quantum cryptographic application does not equate being immune to quantum or traditional cryptanalysis and for many quantum cryptographic application this remains an open question.

[1] https://qt.eu/discover-quantum/underlying-principles/quantum-key-distribution-qkd/. Last accessed May 2021.

BIBLIOGRAPHY

[1] Miklós Ajtai. The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In *30th Annual ACM Symposium on Theory of Computing*, pages 10–19. ACM Press, May 1998.

[2] Martin Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9, 10 2015.

[3] Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang. Classic McEliece. Round 3 submission to NIST post-quantum call for proposals, 2020. https://classic.mceliece.org/.

[4] Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn W. Postlethwaite, Fernando Virdia, and Thomas Wunderer. Estimate all the LWE, NTRU schemes! In Dario Catalano and Roberto De Prisco, editors, *SCN 18: 11th International Conference on Security in Communication Networks*, volume 11035 of *Lecture Notes in Computer Science*, pages 351–367. Springer, Heidelberg, September 2018.

[5] Martin R. Albrecht, Christian Hanser, Andrea Hoeller, Thomas Pöppelmann, Fernando Virdia, and Andreas Wallner. Implementing RLWE-based schemes using an RSA co-processor. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2019(1):169–208, 2018. https://tches.iacr.org/index.php/TCHES/article/view/7338.

[6] Jacob Appelbaum, Chloe Martindale, and Peter Wu. Tiny WireGuard tweak. In Johannes Buchmann, Abderrahmane Nitaj, and Tajje eddine Rachidi, editors, *AFRICACRYPT 19: 11th International Conference on Cryptology in Africa*, volume 11627 of *Lecture Notes in Computer Science*, pages 3–20. Springer, Heidelberg, July 2019.

[7] Nicolas Aragon, Paulo S. L. M. Barreto, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Santosh Ghosh, Shay Gueron, Tim Güneysu, Carlos Aguilar Melchor, Rafael Misoczki, Edoardo Persichetti, Nicolas Sendrier, Jean-Pierre Tillich, Valentin Vasseur, and Gilles Zémor. BIKE - Bit Flipping Key Encapsulation. Round 3 submission to NIST post-quantum call for proposals, 2020. https://bikesuite.org/.

[8] Daniel V. Bailey, Daniel Coffin, Adam J. Elbirt, Joseph H. Silverman, and Adam D. Woodbury. NTRU in constrained devices. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 262–272. Springer, Heidelberg, May 2001.

[9] Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier Verbel. Improvements of algebraic attacks for solving the rank decoding and MinRank problems. In *International Conference on the Theory and Application of Cryptology and Information Security*, 2020.

[10] Andrea Basso, Jose Maria Bermudo Mera, Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Michiel Van Beirendonck, and Frederik Vercauteren. SABER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[11] Michiel Van Beirendonck, Jan-Pieter D'Anvers, Angshuman Karmakar, Josep Balasch, and Ingrid Verbauwhede. A side-channel resistant implementation of saber. Cryptology ePrint Archive, Report 2020/733, 2020. https://eprint.iacr.org/2020/733.

[12] Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsatiansup, Tanja Lange, Adrian Marotzke, Bo-Yuan Peng, Nicola Tuveri, Christine van Vredendaal, and Bo-Yin Yang. NTRU Prime. Round 3 submission to NIST post-quantum call for proposals, 2020. https://ntruprime.cr.yp.to/warnings.html.

[13] Daniel J. Bernstein, Tung Chou, and Peter Schwabe. McBits: Fast constant-time code-based cryptography. In Guido Bertoni and Jean-Sébastien Coron, editors, *Cryptographic Hardware and Embedded Systems – CHES 2013*, volume 8086 of *Lecture Notes in Computer Science*, pages 250–272. Springer, Heidelberg, August 2013.

[14] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. NTRU prime: Reducing attack surface at low cost. In Carlisle Adams and Jan Camenisch, editors, *SAC 2017: 24th Annual International Workshop on Selected Areas in Cryptography*, volume 10719 of *Lecture Notes in Computer Science*, pages 235–260. Springer, Heidelberg, August 2017.

[15] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: Practical stateless hash-based signatures. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 368–397. Springer, Heidelberg, April 2015.

[16] Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. The SPHINCS$^+$ signature framework. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 2129–2146. ACM Press, November 2019.

[17] Daniel J. Bernstein and Tanja Lange. McTiny: Fast high-confidence post-quantum key erasure for tiny network servers. In Srdjan Capkun and Franziska Roesner, editors, *USENIX Security 2020: 29th USENIX Security Symposium*, pages 1731–1748. USENIX Association, August 2020.

[18] Ward Beullens. Improved cryptanalysis of UOV and Rainbow. Cryptology ePrint Archive, Report 2020/1343, 2020. https://eprint.iacr.org/2020/1343.

[19] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 61–90. Springer, Heidelberg, December 2019.

[20] Nina Bindel, Udyani Herath, Matthew McKague, and Douglas Stebila. Transitioning to a quantum-resistant public key infrastructure. In Tanja Lange and Tsuyoshi Takagi, editors, *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017*, pages 384–405. Springer, Heidelberg, 2017.

[21] Nina Bindel and John M. Schanck. Decryption failure is more likely after success. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 206–225. Springer, Heidelberg, 2020.

[22] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, Heidelberg, December 2011.

[23] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In *2018 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 353–367, 2018.

[24] Joppe W. Bos, Joost Renes, and Christine van Vredendaal. Polynomial multiplication with contemporary co-processors: Beyond kronecker, schönhage-strassen and nussbaumer. Cryptology ePrint Archive, Report 2020/1303, 2020. https://eprint.iacr.org/2020/1303.

[25] Leif Both and Alexander May. Optimizing BJMM with nearest neighbors: Full decoding in $2^{2n/21}$ and McEliece security, 2017. International Workshop on Coding and Cryptography (WCC 2017), https://www.cits.ruhr-uni-bochum.de/imperia/md/content/may/paper/bjmm+.pdf.

[26] Leif Both and Alexander May. Decoding linear codes with high error rate and its impact for LPN security. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018*, pages 25–46. Springer, Heidelberg, 2018.

[27] Leon Botros, Matthias J. Kannwischer, and Peter Schwabe. Memory-efficient high-speed implementation of Kyber on cortex-M4. In Johannes Buchmann, Abderrahmane Nitaj, and Tajje eddine Rachidi, editors, *AFRICACRYPT 19: 11th International Conference on Cryptology in Africa*, volume 11627 of *Lecture Notes in Computer Science*, pages 209–228. Springer, Heidelberg, July 2019.

[28] Leon Groot Bruinderink and Peter Pessl. Differential fault attacks on deterministic lattice signatures. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(3):21–43, 2018. https://tches.iacr.org/index.php/TCHES/article/view/7267.

[29] A. Casanova, J.-C. Faugère, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem. GeMSS. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[30] Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, January 2009.

[31] Tung Chou. McBits revisited. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems – CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 213–231. Springer, Heidelberg, September 2017.

[32] David Cooper, Daniel Apon, Quynh Dang, Michael Davidson, Morris Dworkin, and Carl Miller. NIST Special Publication 800-208: Recommendation for Stateful Hash-Based Signature Schemes. Technical report, National Institute of Standards and Technology, 2020. https://doi.org/10.6028/NIST.SP.800-208.

[33] Don Coppersmith and Adi Shamir. Lattice attacks on NTRU. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 52–61. Springer, Heidelberg, May 1997.

[34] Eric Crockett, Christian Paquin, and Douglas Stebila. Prototyping post-quantum and hybrid key exchange and authentication in TLS and SSH. Cryptology ePrint Archive, Report 2019/858, 2019. https://eprint.iacr.org/2019/858.

[35] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 329–358. Springer, Heidelberg, August 2020.

[36] Viet Ba Dang, Farnoud Farahmand, Michal Andrzejczak, Kamyar Mohajerani, Duc Tri Nguyen, and Kris Gaj. Implementation and Benchmarking of Round 2 Candidates in the NIST Post-Quantum Cryptography Standardization Process Using Hardware and Software/Hardware Co-design Approaches. Cryptology ePrint Archive, Report 2020/795, 2020. https://eprint.iacr.org/2020/795.

[37] Jan-Pieter D'Anvers, Qian Guo, Thomas Johansson, Alexander Nilsson, Frederik Vercauteren, and Ingrid Verbauwhede. Decryption failure attacks on IND-CCA secure lattice-based schemes. In Dongdai Lin and Kazue Sako, editors, *PKC 2019: 22nd International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 11443 of *Lecture Notes in Computer Science*, pages 565–598. Springer, Heidelberg, April 2019.

[38] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. In Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *AFRICACRYPT 18: 10th International Conference on Cryptology in Africa*, volume 10831 of *Lecture Notes in Computer Science*, pages 282–305. Springer, Heidelberg, May 2018.

[39] Jan-Pieter D'Anvers, Mélissa Rossi, and Fernando Virdia. (One) failure is not an option: Bootstrapping the search for failures in lattice-based encryption schemes. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part III*, volume 12107 of *Lecture Notes in Computer Science*, pages 3–33. Springer, Heidelberg, May 2020.

[40] Alexander W. Dent. A designer's guide to KEMs. In Kenneth G. Paterson, editor, *9th IMA International Conference on Cryptography and Coding*, volume 2898 of *Lecture Notes in Computer Science*, pages 133–151. Springer, Heidelberg, December 2003.

[41] Jintai Ding, Ming-Shing Chen, Albrecht Petzoldt, Dieter Schmidt, Bo-Yin Yang, Matthias Kannwischer, and Jacques Patarin. Rainbow. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[42] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *ACNS 05: 3rd International Conference on Applied Cryptography and Network Security*, volume 3531 of *Lecture Notes in Computer Science*, pages 164–175. Springer, Heidelberg, June 2005.

[43] Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 602–631. Springer, Heidelberg, August 2020.

[44] Jason A. Donenfeld. WireGuard: Next generation kernel network tunnel. In *ISOC Network and Distributed System Security Symposium – NDSS 2017*. The Internet Society, February / March 2017.

[45] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(1):238–268, 2018. https://tches.iacr.org/index.php/TCHES/article/view/839.

[46] Léo Ducas and Thomas Prest. Fast fourier orthogonalization. In *ISSAC*, pages 191–198. ACM, 2016.

[47] ECRYPT-CSA. Whitepaper on Post-Quantum Cryptography. https://www.ecrypt.eu.org/csa/documents/PQC-whitepaper.pdf, 2018.

[48] Jean-Charles Faugère, Valérie Gauthier, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high rate McEliece cryptosystems. Cryptology ePrint Archive, Report 2010/331, 2010. http://eprint.iacr.org/2010/331.

[49] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, Heidelberg, August 1987.

[50] Pierre-Alain Fouque, Paul Kirchner, Mehdi Tibouchi, Alexandre Wallet, and Yang Yu. Uprooting the Falcon tree? Cryptology ePrint Archive, Report 2019/1180, 2019. https://eprint.iacr.org/2019/1180.

[51] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554. Springer, Heidelberg, August 1999.

[52] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206. ACM Press, May 2008.

[53] T. Güneysu, M. Krausz, T. Oder, and J. Speith. Evaluation of lattice-based signature schemes in embedded systems. In *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 385–388, 2018.

[54] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A new high speed public key cryptosystem, 1996. draft from at CRYPTO '96 rump session. http://web.securityinnovation.com/hubfs/files/ntru-orig.pdf.

[55] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe P. Buhler, editor, *Algorithmic Number Theory – ANTS-III*, volume 1423 of *LNCS*, pages 267–288. Springer, 1998. http://dx.doi.org/10.1007/BFb0054868.

[56] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371. Springer, Heidelberg, November 2017.

[57] James Howe, Thomas Prest, Thomas Ricosset, and Mélissa Rossi. Isochronous gaussian sampling: From inception to implementation. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 53–71. Springer, Heidelberg, 2020.

[58] Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 150–169. Springer, Heidelberg, August 2007.

[59] Andreas Hülsing, Denise Butin, Stefan-Lukas Gazdag, Joost Rijneveld, and Aziz Mohaisen. XMSS: Extended Hash-Based Signatures. Internet Requests for Comments, 2018.

[60] Andreas Hülsing, Joost Rijneveld, John M. Schanck, and Peter Schwabe. High-speed key encapsulation from NTRU. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems – CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 232–252. Springer, Heidelberg, September 2017.

[61] Andreas Hülsing, Kai-Chun Ning, Peter Schwabe, Florian Weber, and Philip R. Zimmermann. Post-quantum WireGuard. Cryptology ePrint Archive, Report 2020/379, 2020. https://eprint.iacr.org/2020/379.

[62] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th Annual ACM Symposium on Theory of Computing*, pages 21–30. ACM Press, June 2007.

[63] David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, and Geovandro Pereira. SIKE. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[64] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 96–125. Springer, Heidelberg, August 2018.

[65] Daniel Kales and Greg Zaverucha. Improving the Performance of the Picnic Signature Scheme. Cryptology ePrint Archive, Report 2020/427, 2020. https://eprint.iacr.org/2020/427.

[66] Matthias J. Kannwischer, Joost Rijneveld, and Peter Schwabe. Faster multiplication in $\mathbb{Z}_{2^m}[x]$ on cortex-M4 to speed up NIST PQC candidates. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 19: 17th International Conference on Applied Cryptography and Network Security*, volume 11464 of *Lecture Notes in Computer Science*, pages 281–301. Springer, Heidelberg, June 2019.

[67] Matthias J. Kannwischer, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. pqm4: Testing and benchmarking NIST PQC on ARM cortex-M4. Cryptology ePrint Archive, Report 2019/844, 2019. https://eprint.iacr.org/2019/844.

[68] Angshuman Karmakar, Jose Maria Bermudo Mera, Sujoy Sinha Roy, and Ingrid Verbauwhede. Saber on ARM. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(3):243–266, 2018. https://tches.iacr.org/index.php/TCHES/article/view/7275.

[69] Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. Pushing the speed limit of constant-time discrete Gaussian sampling. A case study on the Falcon signature scheme. In *DAC*, page 88. ACM, 2019.

[70] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020.

[71] Elena Kirshanova. Improved quantum information set decoding. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018*, pages 507–527. Springer, Heidelberg, 2018.

[72] Stefan Kölbl, Martin M. Lauridsen, Florian Mendel, and Christian Rechberger. Haraka v2 - Efficient short-input hashing for post-quantum applications. *IACR Transactions on Symmetric Cryptology*, 2016(2):1–29, 2016. http://tosc.iacr.org/index.php/ToSC/article/view/563.

[73] Juliane Krämer and Mirjam Loiero. Fault attacks on UOV and Rainbow. In Ilia Polian and Marc Stöttinger, editors, *COSADE 2019: 10th International Workshop on Constructive Side-Channel Analysis and Secure Design*, volume 11421 of *Lecture Notes in Computer Science*, pages 193–214. Springer, Heidelberg, April 2019.

[74] Leslie Lamport. Constructing digital signatures from a one-way function. Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, October 1979.

[75] Tanja Lange. SD8 (Post-Quantum Cryptography) – Part 4: Code-Based Cryptography. Technical Report N 2276, ISO/IEC JTC 1/SC 27/WG 2, 2020. https://www.din.de/resource/blob/721042/4f1941ac1de9685115cf53bc1a14ac61/sc27wg2-sd8-data.zip.

[76] Tanja Lange. Sd8 (post-quantum cryptography) – part 6: Isogeny-based cryptography. Technical Report N 2274, ISO/IEC JTC 1/SC 27/WG 2, 2020. https://www.din.de/resource/blob/721042/4f1941ac1de9685115cf53bc1a14ac61/sc27wg2-sd8-data.zip.

[77] Adam Langley. CECPQ2, 2018. https://www.imperialviolet.org/2018/12/12/cecpq2.html.

[78] Mun-Kyu Lee, Jeong Eun Song, Dooho choi, and Dong-Guk Han. Countermeasures against power analysis attacks for the NTRU public key cryptosystem. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E93.A(1):153–163, 2010.

[79] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, Heidelberg, February 2011.

[80] Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[81] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, Heidelberg, May / June 2010.

[82] J. Maria Bermudo Mera, F. Turan, A. Karmakar, S. Sinha Roy, and I. Verbauwhede. Compact domain-specific co-processor for accelerating module lattice-based kem. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2020.

[83] Alexander May. Cryptanalysis of NTRU, 1999. https://www.cits.ruhr-uni-bochum.de/imperia/md/content/may/paper/cryptanalysisofntru.ps.

[84] Alexander May and Joseph H. Silverman. Dimension reduction methods for convolution modular lattices. In Joseph H. Silverman, editor, *Cryptography and Lattices: International Conference – CaLC 2001*, volume 2146 of *LNCS*, pages 110–125. Springer, 2001. http://dx.doi.org/10.1007/3-540-44670-2_10.

[85] Sarah McCarthy, James Howe, Neil Smyth, Séamus Brannigan, and Máire O'Neill. BEARZ attack FALCON: implementation attacks with countermeasures on the FALCON signature scheme. In *ICETE (2)*, pages 61–71. SciTePress, 2019.

[86] Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory, 1978. JPL DSN Progress Report http://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF.

[87] David A. McGrew, Michael Curcio, and Scott R. Fluhrer. Hash-Based Signatures. RFC 8554, RFC Editor, 2019.

[88] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jurjen Bos, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, Jean-Marc Robert, Pascal Véron, and Gilles Zémor. HQC (Hamming Quasi-Cyclic). Round 3 submission to NIST post-quantum call for proposals, 2020. http://pqc-hqc.org/.

[89] Jose Maria Bermudo Mera, Angshuman Karmakar, and Ingrid Verbauwhede. Time-memory trade-off in Toom-Cook multiplication. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(2):222–244, 2020. https://tches.iacr.org/index.php/TCHES/article/view/8550.

[90] Ralph C. Merkle. A certified digital signature. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 218–238. Springer, Heidelberg, August 1990.

[91] Vincent Migliore, Benoît Gérard, Mehdi Tibouchi, and Pierre-Alain Fouque. Masking Dilithium - efficient implementation and side-channel evaluation. In Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung, editors, *ACNS 19: 17th International Conference on Applied Cryptography and Network Security*, volume 11464 of *Lecture Notes in Computer Science*, pages 344–362. Springer, Heidelberg, June 2019.

[92] Michael Naehrig, Erdem Alkim, Joppe Bos, Léo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila. FrodoKEM. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[93] Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15(2):159–166, 1986.

[94] Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, 2016. https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[95] Status report on the second round of the nist post-quantum cryptography standardization process, 2020. https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf.

[96] Tobias Oder, Tobias Schneider, Thomas Pöppelmann, and Tim Güneysu. Practical CCA2-secure masked Ring-LWE implementations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(1):142–174, 2018. https://tches.iacr.org/index.php/TCHES/article/view/836.

[97] Tobias Oder, Julian Speith, Kira Höltgen, and Tim Güneysu. Towards practical microcontroller implementation of the signature scheme Falcon. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 65–80. Springer, Heidelberg, 2019.

[98] Irem Keskinkurt Paksoy and Murat Cenk. TMVP-based Multiplication for Polynomial Quotient Rings and Application to Saber on ARM Cortex-M4. Cryptology ePrint Archive, Report 2020/1302, 2020. https://eprint.iacr.org/2020/1302.

[99] Jacques Patarin. The oil and vinegar signature scheme. In *Dagstuhl Workshop on Cryptography September, 1997*, 1997.

[100] Trevor Perrin. Noise protocol framework, 2018. https://noiseprotocol.org/noise.pdf, Revision 34, 2018-07-11.

[101] Thomas Pornin. New efficient, constant-time implementations of Falcon. Cryptology ePrint Archive, Report 2019/893, 2019. https://eprint.iacr.org/2019/893.

[102] E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions*, IT-8:S5–S9, 1962.

[103] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[104] Prasanna Ravi, Sourav Sen Gupta, Anupam Chattopadhyay, and Shivam Bhasin. Improving speed of Dilithium's signing procedure. Cryptology ePrint Archive, Report 2019/420, 2019. https://eprint.iacr.org/2019/420.

[105] Prasanna Ravi, Debapriya Basu Roy, Shivam Bhasin, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Number "not used" once - practical fault attack on pqm4 implementations of NIST candidates. In Ilia Polian and Marc Stöttinger, editors, *COSADE 2019: 10th International Workshop on Constructive Side-Channel Analysis and Secure Design*, volume 11421 of *Lecture Notes in Computer Science*, pages 232–250. Springer, Heidelberg, April 2019.

[106] Sujoy Sinha Roy and Andrea Basso. High-speed instruction-set coprocessor for lattice-based key encapsulation mechanism: Saber in hardware. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(4):443–466, 2020. https://tches.iacr.org/index.php/TCHES/article/view/8690.

[107] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 520–551. Springer, Heidelberg, April / May 2018.

[108] Simona Samardjiska, Ming-Shing Chen, Andreas Hulsing, Joost Rijneveld, and Peter Schwabe. MQDSS. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[109] Thomas Schamberger, Oliver Mischke, and Johanna Sepúlveda. Practical evaluation of masking for NTRUEncrypt on ARM cortex-M4. In Ilia Polian and Marc Stöttinger, editors, *COSADE 2019: 10th International Workshop on Constructive Side-Channel Analysis and Secure Design*, volume 11421 of *Lecture Notes in Computer Science*, pages 253–269. Springer, Heidelberg, April 2019.

[110] John M. Schanck, Andreas Hulsing, Joost Rijneveld, and Peter Schwabe. NTRU-HRSS-KEM. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[111] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[112] S. Sinha Roy. SaberX4: High-Throughput Software Implementation of Saber Key Encapsulation Mechanism. In *2019 IEEE 37th International Conference on Computer Design (ICCD)*, pages 321–324, 2019.

[113] Daniel Smith-Tone and Ray Perlner. Rainbow band separation is better than we thought. Cryptology ePrint Archive, Report 2020/702, 2020. https://eprint.iacr.org/2020/702.

[114] Bin Wang, Xiaozhuo Gu, and Yingshan Yang. Saber on ESP32. In Mauro Conti, Jianying Zhou, Emiliano Casalicchio, and Angelo Spognardi, editors, *ACNS 20: 18th International Conference on Applied Cryptography and Network Security, Part I*, volume 12146 of *Lecture Notes in Computer Science*, pages 421–440. Springer, Heidelberg, October 2020.

[115] Wen Wang, Jakub Szefer, and Ruben Niederhagen. FPGA-based key generator for the niederreiter cryptosystem using binary goppa codes. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems – CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 253–274. Springer, Heidelberg, September 2017.

[116] Wen Wang, Jakub Szefer, and Ruben Niederhagen. FPGA-based niederreiter cryptosystem using binary goppa codes. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018*, pages 77–98. Springer, Heidelberg, 2018.

[117] Zhenfei Zhang, Cong Chen, Jeffrey Hoffstein, and William Whyte. NTRUEncrypt. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

[118] Zhenfei Zhang, Cong Chen, Jeffrey Hoffstein, William Whyte, John M. Schanck, Andreas Hulsing, Joost Rijneveld, Peter Schwabe, and Oussama Danba. NTRUEncrypt. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

[119] Yihong Zhu, Min Zhu, Bohan Yang, Wenping Zhu, Chenchen Deng, Chen Chen, Shaojun Wei, and Leibo Liu. A high-performance hardware implementation of saber based on karatsuba algorithm. Cryptology ePrint Archive, Report 2020/1037, 2020. https://eprint.iacr.org/2020/1037.

[120] Philip Zimmerman, Alan Johnston, and Jon Callas. ZRTP: Media Path Key Agreement for Unicast Secure RTP. RFC 6189, RFC Editor, 2011.

ABOUT ENISA

The European Union Agency for Cybersecurity, ENISA, is the Union's agency dedicated to achieving a high common level of cybersecurity across Europe. Established in 2004 and strengthened by the EU Cybersecurity Act, the European Union Agency for Cybersecurity contributes to EU cyber policy, enhances the trustworthiness of ICT products, services and processes with cybersecurity certification schemes, cooperates with Member States and EU bodies, and helps Europe prepare for the cyber challenges of tomorrow. Through knowledge sharing, capacity building and awareness raising, the Agency works together with its key stakeholders to strengthen trust in the connected economy, to boost resilience of the Union's infrastructure, and, ultimately, to keep Europe's society and citizens digitally secure. More information about ENISA and its work can be found here: www.enisa.europa.eu.

ENISA
European Union Agency for Cybersecurity

Athens Office
1 Vasilissis Sofias Str
151 24 Marousi, Attiki, Greece

Heraklion office
95 Nikolaou Plastira
700 13 Vassilika Vouton, Heraklion, Greece

enisa.europa.eu

ISBN 978-92-9204-468-8
DOI 10.2824/92307